



P6192a

COPY OF PAPERS
ORIGINALLY FILED2122#4
5-16-02
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventors: Junichi Otsuka, et al.

Group Art Unit: 2122

Serial No.: 10/026,336

Examiner: Not Yet Assigned

Filed: December 20, 2001

Title: CHARACTER STRING PROCESSING METHOD, PROCESSING SYSTEM, AND
OBJECT PROGRAM THEREFOR

CERTIFICATE OF MAILING

I hereby certify that this correspondence and the documents referred to as attached herein are being deposited with the United States Postal Service on this date in an envelope as "First Class Mail" service addressed to the Assistant Commissioner for Patents, Washington, DC 20231.

Date: February 21, 2002

Virginia Silva

SUBMISSION OF PRIORITY DOCUMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Enclosed is the certified copy of the Japanese patent application listed below. The claim of priority under 35 USC §119 in the above-identified application is based on this Japanese patent application.

Japanese Patent Application

<u>Number</u>	<u>Date Filed</u>
2000-40330G	December 28, 2000

Respectfully submitted,

Michael T. Gabrik
Attorney for Applicants
Registration No. 32,896

Please address all correspondence to:
Epson Research and Development, Inc.
Intellectual Property Department
150 River Oaks Parkway, Suite 225
San Jose, CA 95134
Customer No. 20178
Phone: (408) 952-6000
Fax: (408) 954-9058
Date: February 21, 2002

RECEIVED

MAR 13 2002

Technology Center 2100



日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年12月28日

出 願 番 号

Application Number:

特願2000-403300

[ST.10/C]:

[JP2000-403300]

出 願 人

Applicant(s):

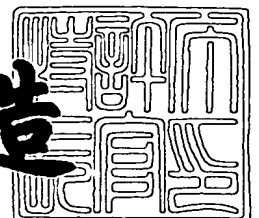
セイコーエプソン株式会社

RECEIVED
MAR 13 2002
Technology Center 2100

2002年 1月25日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2002-3001434

【書類名】 特許願

【整理番号】 J0082710

【提出日】 平成12年12月28日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 13/10

【発明者】

 【住所又は居所】 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

 【氏名】 大塚 順一

【発明者】

 【住所又は居所】 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

 【氏名】 杉本 季之

【特許出願人】

 【識別番号】 000002369

 【氏名又は名称】 セイコーエプソン株式会社

【代理人】

 【識別番号】 100093388

 【弁理士】

 【氏名又は名称】 鈴木 喜三郎

 【連絡先】 0266-52-3139

【選任した代理人】

 【識別番号】 100095728

 【弁理士】

 【氏名又は名称】 上柳 雅誉

【選任した代理人】

 【識別番号】 100107261

 【弁理士】

 【氏名又は名称】 須澤 修

【手数料の表示】

【予納台帳番号】 013044

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9711684

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 文字列データの処理方法および処理装置

【特許請求の範囲】

【請求項 1】 アプリケーションプログラムを実行する際に、クライアント側となる第 1 のオブジェクトからサーバ側となる第 2 のオブジェクトに対し、コンピュータのオペレーティングシステム上で稼動するインターフェイスオブジェクトを介して第 1 の文字列データを転送する際に、

前記インターフェイスオブジェクトにおいて、前記第 1 のオブジェクトの前記第 1 の文字列データの宣言された型に関わらず前記第 1 の文字列データを、文字列データ中に含まれる `null` データも含めて文字数をカウントできる基本文字列データとして前記第 1 の文字列データの実効文字数を求め、前記オペレーティングシステムが転送可能な転送文字列データに前記第 1 の文字列データを前記実効文字数だけ変換する工程と、

前記第 2 のオブジェクトにおいて、前記第 1 の文字列データの宣言された型に関わらず、前記第 1 の文字列データを前記基本文字列データとして処理を実行する工程とを有する文字列データの処理方法。

【請求項 2】 請求項 1 において、前記基本文字列データは、文字数データ付きの文字列データである文字列データの処理方法。

【請求項 3】 請求項 1 において、前記アプリケーションプログラムは、`null` データを含む文字列データを処理可能であり、前記第 1 のオブジェクトは、前記アプリケーションプログラムまたは文字列データを処理可能な周辺装置の種類単位のコントロールオブジェクトであり、前記第 2 のオブジェクトは、前記制御オブジェクトまたは前記周辺装置毎のサービスオブジェクトである文字列データの処理方法。

【請求項 4】 請求項 3 において、前記文字列データはバーコードデータである文字列データの処理方法。

【請求項 5】 請求項 3 において、前記アプリケーションプログラムは P O S アプリケーションプログラムである文字列データの処理方法。

【請求項 6】 アプリケーションプログラムを実行する際に、クライアント

側となる第1のオブジェクトと、この第1のオブジェクトに対しサーバ側となる第2のオブジェクトと、前記第1のオブジェクトから第2のオブジェクトにオペレーティングシステムを介して第1の文字列データを転送可能なインターフェイスオブジェクトとを有し、

前記インターフェイスオブジェクトは、前記第1のオブジェクトの前記第1の文字列データの宣言された型に関わらず前記第1の文字列データを、文字列データ中に含まれる `null` データも含めて文字数をカウントできる基本文字列データとして前記第1の文字列データの実効文字数を求め、前記オペレーティングシステムが転送可能な転送文字列データに前記第1の文字列データを前記実効文字数だけ変換する手段を備えており、

前記第2のオブジェクトは、前記第1の文字列データの宣言された型に関わらず、前記第1の文字列データを前記基本文字列データとして処理を実行する手段を備えている文字列データ処理システム。

【請求項7】 請求項6において、前記基本文字列データは、文字数データ付きの文字列データである文字列データ処理システム。

【請求項8】 請求項6において、前記アプリケーションプログラムは、`null` データを含む文字列データを処理可能であり、前記第1のオブジェクトは、前記アプリケーションプログラムまたは文字列データを処理可能な周辺装置の種類単位のコントロールオブジェクトであり、前記第2のオブジェクトは、前記制御オブジェクトまたは前記周辺装置毎のサービスオブジェクトである文字列データ処理システム。

【請求項9】 請求項8において、前記文字列データはバーコードデータである文字列データ処理システム。

【請求項10】 請求項8において、前記アプリケーションプログラムはP O Sアプリケーションプログラムである文字列データ処理システム。

【請求項11】 アプリケーションプログラムを実行する際に、クライアント側となるオブジェクトプログラムであって、

当該オブジェクトに対しサーバ側となるオブジェクトに対し、オペレーティングシステムを介して第1の文字列データを転送可能なインターフェイスオブジェ

クトを呼び出して前記第1の文字列データを転送する工程を有し、この工程において、当該オブジェクトの前記第1の文字列データの宣言された型に関わらず前記第1の文字列データを、文字列データ中に含まれるnullデータも含めて文字数をカウントできる基本文字列データとして前記第1の文字列データの実効文字数を求め、前記オペレーティングシステムが転送可能な転送文字列データに前記第1の文字列データを前記実効文字数だけ変換する前記インターフェイスオブジェクトを呼び出す処理を実行可能な命令を有するオブジェクトプログラム。

【請求項12】 請求項11において、前記オブジェクトプログラムは、nullデータを含む文字列データを処理可能なアプリケーションプログラム、または、文字列データを処理可能な周辺装置の種類単位のコントロールオブジェクトであることを特徴とするオブジェクトプログラム。

【請求項13】 請求項11において、前記文字列データはバーコードデータであり、前記アプリケーションプログラムはPOSアプリケーションプログラムであるオブジェクトプログラム。

【請求項14】 アプリケーションプログラムを実行する際に、サーバ側となるオブジェクトプログラムであって、当該オブジェクトに対しクライアント側となるオブジェクトから、オペレーティングシステム上で稼動するインターフェイスオブジェクトを介して転送された第1の文字列データを、当該オブジェクトにおける前記第1の文字列データの宣言された型に関わらず、文字列データ中に含まれるnullデータも含めて文字数をカウントできる基本文字列データとして処理を実行可能な命令を有するオブジェクトプログラム。

【請求項15】 請求項14において、前記オブジェクトプログラムは、文字列データを処理可能な周辺装置の種類単位のコントロールオブジェクト、または前記周辺装置毎のサービスオブジェクトであることを特徴とするオブジェクトプログラム。

【請求項16】 請求項14において、前記文字列データはバーコードデータであり、前記アプリケーションプログラムはPOSアプリケーションプログラムであるオブジェクトプログラム。

【請求項17】 請求項11ないし16のいずれかに記載のオブジェクトプ

ログラムが記録されていることを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 1 8】 アプリケーションプログラムを実行する際に、クライアント側となる第 1 のオブジェクトからサーバ側となる第 2 のオブジェクトに対し、コンピュータのオペレーティングシステムを介して第 1 の文字列を転送可能なインターフェイスオブジェクトプログラムであって、

前記第 1 のオブジェクトの前記第 1 の文字列データの宣言された型に関わらず前記第 1 の文字列データを、文字列データ中に含まれる `null` データも含めて文字数をカウントできる基本文字列データとして前記第 1 の文字列データの実効文字数を求め、前記オペレーティングシステムが転送可能な転送文字列データに前記第 1 の文字列データを前記実効文字数だけ変換する処理を実行可能な命令を有するインターフェイスオブジェクトプログラム。

【請求項 1 9】 請求項 1 8 において、前記基本文字列データは、文字数データ付きの文字列データであるインターフェイスオブジェクトプログラム。

【請求項 2 0】 請求項 1 8 または 1 9 に記載のインターフェイスオブジェクトプログラムが記録されていることを特徴とするコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、コンポーネント化されたオブジェクトソフトウェアを用いた処理方法およびシステムに関するものである。

【0 0 0 2】

【従来の技術】

オブジェクト指向プログラミングによるコンポーネントベースのプログラムあるいはソフトウェア（オブジェクト）は、オペレーティングシステム上で稼動する再利用可能なソフトウェアであり、オブジェクトを用いて多種多様なシステムが開発されている。たとえば、マイクロソフト社が提供するオペレーティングシステム（OS）であるウィンドウズ（登録商標）でサポートする COM（コンポ

ーネットオブジェクトモデル：Component Object Model) は、クライアントであるアプリケーションプログラム（コンテナアプリケーション）がオブジェクトを使用可能とする環境を提供する広範囲な機能を有するシステムである。

【 0 0 0 3 】

したがって、COMで利用可能なオブジェクトによりシステム開発を行うと、ソフトウェアの部品化と再利用が容易になり、ユーザーがシステムを構築し、メンテナンスしていく費用を低減できる。また、既存のソフトウェア資産を継続あるいは承継できるので、システム開発も容易となり、それに伴う費用も削減できる。さらに、コンポーネントベースでオブジェクトが開発できるので、多種多様なサブシステムあるいは周辺装置により構成される複雑な構成のシステムの開発も容易になり、ネットワーク環境下で稼動するシステムの開発には欠かせないものとなっている。

【 0 0 0 4 】

POSシステムは多種多様な周辺装置により組み立てられるシステムの1つであり、パーソナルコンピュータなどの中心となるホストコンピュータにディスプレイ、プリンタ、スキャナ、ドロワ、カードリーダーなど様々な入出力装置が接続されることによりシステムが構築される。本願出願人は、特開平9-106355号などにより、オブジェクトを用い構築されたPOSシステムであってパーソナルコンピュータベースのフレキシブルな構成のPOSシステムについて開示している。

【 0 0 0 5 】

図1に、オブジェクト分散型のPOSシステム10の概要を示してある。このPOSシステム10は、各メーカーが提供するPOSアプリケーションプログラム11と、プリンタ21、スキャナ22あるいはドロワ23などの周辺装置（デバイス）の種類単位（デバイスクラス単位）に供給されるコントロールオブジェクト12と、周辺装置毎、すなわち、デバイス毎に供給されるサービスオブジェクト13とを備えている。したがって、POSアプリケーションプログラム11がプリンタ21から出力するときは、データがプリンタのコントロールオブジェクト（CO）12に渡され、さらに、出力するプリンタのメーカーおよび機種に対応

したサービスオブジェクト（SO）13に渡され、OS14を介してプリンタ21に供給される。また、POSアプリケーションプログラム11は、OS14を介してパーソナルコンピュータのハードウェア資源であるディスプレイ15やキーボード16も周辺装置として使用できるようになっている。

【0006】

図1に示したPOSシステム10では、コントロールオブジェクト12およびサービスオブジェクト13を備えたオブジェクト制御システム19がPOSアプリケーションプログラム11に対してプリンタ21などの周辺機器に対して機種依存性のないインターフェイスを提供する。また、コントロールオブジェクト13は周辺機器のデバイスクラス毎に開発されているので、その範囲でプログラム11を実行するときOS14にサポートされているCOMによりインスタンスが形成され、プログラム11にバインドされる。さらに、サービスオブジェクト13はデバイス毎にコントロールオブジェクト12に対してデバイスに依存しないユニホームなインターフェイスを提供しており、対応するデバイスがコントロールオブジェクト12を介して呼ばれるとインスタンスが形成されてバインドされる。たとえば、マイクロソフト社のCOMを基盤とするシステムであれば、コントロールオブジェクト12としてはプログラム11にイベントを返すことが可能なActiveXコントロール（OLEコントロール）を採用でき、サービスオブジェクト13としてはActiveXコントロールに対してサーバとなるActiveXサーバ（OLEオートメーションサーバ）を採用できる。そして、オートメーションによりバインディングされたこれらのオブジェクト間、すなわちアプリケーションを実行するためにバインドされたオブジェクトのインスタンス間のデータ通信をCOMがサポートしている。

【0007】

オートメーションはCOMオブジェクトのIDispatchインターフェイスに依存しており、オブジェクト間の通信はInvokeというメソッドを実行することにより行われる。このため、オブジェクト指向のプログラミング言語であるマイクロソフト社が提供するVisual C++（以下VC++）（登録商標）によるOLEオートメーション機能を用いたシステムの開発環境を提供するMF

C (Microsoft Foundation Class Library) では、アプリケーションあるいはクライアント側のオブジェクトから `Invoke` 関数をコールする複雑さをカバーするために `COleDispatchDriver` というベースクラス (オブジェクト) が用意されている。したがって、クライアント側のオブジェクトは、このクラスから派生させたインターフェイスオブジェクトのメンバー関数である `InvokeHelper` をコールすることにより、サーバ側のオブジェクトに対しデータを転送することができる。

【 0 0 0 8 】

図 2 に、本例の POS システム 1 0 において、データが転送される様子を模式的に示してある。まず、アプリケーション 1 1 はプリンタ 2 1 で文字データを印刷するために、プリンタのコントロールオブジェクト 1 2 の印刷を行うためのメソッドを呼んで文字データを渡す。コントロールオブジェクト 1 2 は、プリンタ 2 1 に対応したサービスオブジェクト 1 3 のメソッドを呼ぶために、インターフェイスオブジェクト 1 8 である `COleDispatchDriver` クラスのメンバー関数 `InvokeHelper` を呼ぶ。そして、このインターフェイスオブジェクト 1 8 を介してデータがサービスオブジェクト 1 3 に転送され、OS 1 4 を介してプリンタ 2 1 により出力される。

【 0 0 0 9 】

【発明が解決しようとする課題】

しかしながら、この POS システム 1 0 において、バーコードデータのように `null` データ (16 進「00」、以降においては文字列データは全て 16 進で示す) が文字列データの途中に含まれる可能性がある、その文字列がプリンタ 2 1 に再現されないという問題がある。たとえば、プリンタのコントロールオブジェクト 1 2 は `PrintBarCode` というメソッドを備えているが、その `Data` パラメータに以下のデータ (16 進表記) を指定した場合に次のようなプロセスでデータが受け渡しされてしまう。

【 0 0 1 0 】

アプリケーション 1 1 で次の文字列データ A 0 がコントロールオブジェクト 1 2 に渡される。

A 0 : 7B 41 00 30 31 32 33 34 35 36 37 38 39 . . . (1)

コントロールオブジェクト 12 では、文字列データ A0 を受け取るが、受け取る文字列データの形式は、LPCTSTR型で宣言されている。LPCTSTR型は、文字列の終端が「00」で定義された8バイトのANSI規格によるコードであり、さらにシフトJISで文字が定義される文字列データである。したがって、文字列中に「00」があることは許容されていない。バーコードのようにキャラクタではなくバイトデータを画像データとして印刷しようとする場合は、印刷可能なキャラクタに「00」は含まれておらず、プリンタ用のオブジェクトにおいては文字列データの定義としてLPCTSTR型は一般的なものである。

【 0 0 1 1 】

一方、COMの環境では、COMがプログラム-オブジェクト間およびオブジェクト-オブジェクト間のデータ転送をサポートしており、COMを介して文字列データを受け取るときは文字数データ付きの基本文字列データであるBSTR型で渡されるので、以下のようなデータとなる。

A 1 : 0D 00 00 00 7B 41 00 30 31 32 33 34 35 36 37 38 39 00 . . . (2)

BSTR型の文字列データは、先頭の4バイトにはデータの長さの情報が格納され、文字列データの最後（文字数には含まれない）が「00」で終了するデータである。そして、ポインタからの文字数によりデータ領域が管理されるので、文字列データ中に「00」があっても文字列データとして取り扱われる。

【0 0 1 2】

ここで、コントロールオブジェクト12がサービスオブジェクト13のプリント出力用のメソッドを呼ぶために、MFCのライブラリとして用意されているインターフェイスオブジェクト18であるCOleDispatchDriverクラスのメンバ関数InvokeHelperを呼び、InvokeHelperはさらに、InvokeHelperVを以下のように呼ぶ。

```
void COleDispatchDriver::InvokeHelperV(DISPID dwDispID, WORD wFlags, VAR
TYPE vtRet, Void* pvRet, const BYTE*, pbParamInfo, va_list argList)
```

. . . (3)

引数は、それぞれ、dwDispIDが呼び出されるメソッドまたはプロパティを識別し、wFlagsがIDispatch::Invokeの呼び出しコンテキストを示すフラグであり、vtRetが戻り値の型を指定し、pvRetが戻り値またはプロパティ値を受け取る変数のアドレスを指定し、pbParamInfoがこれに続く引数の型を指定するバイト列のNULLで終わる文字列へのポインタを指定し、argListが引数のリストを指定する。

【 0 0 1 3 】

これにより、COleDispatchDriver::InvokeHelperVは、文字列データを以下のように変換する。

【 0 0 1 4 】

```
LPCSTR lpsz = va_arg(argList, LPSTR);
pArg->bstrVal = ::SysAllocString(T2COLE(lpsz));
if (lpsz != NULL && pArg->bstrVal == NULL)
    AfxThrowMemoryException();
pArg->vt = VT_BSTR; . . . (4)
```

ここでは、コントロール12から渡された文字列データlpszが8ビット（マルチバイト）のシフトJIS（以降ではTキャラクタ）であるLPCTSTR型で宣言されているので、文字変換マクロT2COLEが自動的に選択され、COMを介してデータ転送する基本的な文字列（転送文字列）である16ビット（ワイドキャラクタ）のUNICODEに変換する。まず、最初の行では、引数のリストからDataパラメータを取り出し、次の行で実際にデータの変換が行われる。文字変換マクロT2COLEでは、AfxA2WHelperが呼ばれ、その関数内のMultiByteToWideCharにより、データは以下の16ビット単位のワイドキャラクタのデータに変換される。

A 2 : 7B 00 41 00 00 00 . . . (5)

すなわち、コントロールオブジェクトで文字列データがnullデータを終端とするLPCTSTR型で宣言されているので、文字列データ中に含まれている

「00」を文字列データの終端として変換される。このため、「00」以降のデータを含まない、「00」以降が削除された、終端が「00」のワイドキャラクタ (LPCOLESTR型) の文字列データとなる。そして、SysAllocStringにより、文字数 (実際にはバイト数) データを備えたBSTR型に変換されCOMを介してサービスオブジェクト13に転送される。

【0015】

すなわち、サービスオブジェクト13に転送されるデータは4バイトの文字数データが加わり以下のようなになる。

A3: 04 00 00 00 7B 00 41 00 . . . (6)

サービスオブジェクト13では、コントロールオブジェクト12と同様に文字列データがLPCTSTR型で宣言されており、COMにより、宣言された型の文字列データに合わせた8ビットのデータを受け取る。すなわち、サービスオブジェクトにおいては以下のような文字列データを受け取る。

A4: 7B 41 00 . . . (7)

サービスオブジェクト13により制御されるプリンタ21では、アプリケーションが指定したデータを印字することができない。COMにより送られる文字列データはBSTR型であり、上記の文字列データA4をBSTR型でキャストして処理することも可能であるが、メモリを見るとデータは以下のようにっており、BSTR型としての文字列の長さの情報を持っているが、コントロールオブジェクトから転送されるときに切り捨てられたデータは残されていない。

A5: 02 00 00 00 7B 41 00 . . . (8)

このように、コントロールオブジェクト12においてLPCTSTR型で宣言した文字列データでアプリケーションプログラム11から「00」を含むデータを受信すると、それをサービスオブジェクト13に転送してプリンタ21で出力することができない。一方、コントロールオブジェクトやサービスオブジェクトにおいて文字列データの宣言された型を変更することは、オブジェクト間、あるいはアプリケーションとオブジェクト間で交換されるパラメータの仕様の統一を図り、ユニホームなインターフェイスをアプリケーションプログラムに提供するオブジェクト指向のPOSシステムでは採用することができない。また、ユーザ

がすでに導入しているソフトウェア資産を保護するという点でも好ましくない。

【0016】

また、バイナリコンバージョンを利用して「00」を強制的に他の値に変換してから転送し、次のオブジェクト内で元に戻すという方法がないわけではない。しかしながら、処理が複雑になることはもちろんであるが、オブジェクト内で、他のオブジェクトの変換処理とマッチした処理を行う必要があり、転送元のオブジェクトの仕様に依存することになる。したがって、ソフトウェアの部品化と再利用の可能性を高めるというオブジェクト指向プログラムとは相反する。さらに、バイナリコンバージョンされたデータを処理できないオブジェクトであれば、異なった文字列を出力する可能性がある。

【0017】

このように「00」を含む文字列データの処理は難しい問題を含んでいるが、バーコードを含め、多種多様なデータが取り扱えるPOSシステムを構築するためには「00」を含む文字列データがサービスオブジェクトに正確に転送され、プリンタなどのデバイスで出力できるようにすることが必要である。

【0018】

そこで、本発明においては、コントロールオブジェクトあるいはサービスオブジェクトの文字列データの型の宣言を変更したり、複雑な処理を行わないで、「00」すなわちnullデータを含む文字列データであっても、正確にオブジェクト間で転送することができる文字列データの処理方法、処理システムおよびそれに用いられるオブジェクトプログラムを提供することを目的としている。

【0019】

【課題を解決するための手段】

このため、本発明においては、アプリケーションプログラムを実行する際に、クライアント側となる第1のオブジェクト、たとえば上記のコントロールオブジェクトからサーバ側となる第2のオブジェクト、たとえば上記のサービスオブジェクトに対し、COMなどのコンピュータのオペレーティングシステム上で稼動するインターフェイスオブジェクトを介して第1の文字列データを転送する際に、インターフェイスオブジェクトにおいて、第1のオブジェクトの第1の文字列

データの宣言された型に関わらず第1の文字列データを、文字列データ中に含まれる `null` データも含めて文字数をカウントできる基本文字列データとして第1の文字列データの実効文字数を求め、オペレーティングシステムが転送可能な転送文字列データに第1の文字列データを実効文字数だけ変換する工程と、第2のオブジェクトにおいて、第1の文字列データの宣言された型に関わらず、第1の文字列データを基本文字列データとして処理を実行する工程とを有する文字列データの処理方法により処理するようにしている。

【0020】

したがって、本発明の、アプリケーションプログラムを実行する際に、クライアント側となる第1のオブジェクトと、この第1のオブジェクトに対しサーバ側となる第2のオブジェクトと、第1のオブジェクトから第2のオブジェクトにオペレーティングシステムを介して第1の文字列データを転送可能なインターフェイスオブジェクトとを有する文字列データ処理システムにおいては、インターフェイスオブジェクトは、第1のオブジェクトの第1の文字列データの宣言された型に関わらず第1の文字列データを、文字列データ中に含まれる `null` データも含めて文字数をカウントできる基本文字列データとして第1の文字列データの実効文字数を求め、オペレーティングシステムが転送可能な転送文字列データに第1の文字列データを前記実効文字数だけ変換する手段を備えている。さらに、第2のオブジェクトは、第1の文字列データの宣言された型に関わらず、第1の文字列データを基本文字列データとして処理を実行する手段を備えている。

【0021】

オブジェクトで宣言されている型に関わらず、文字列データを、その文字列データ中に含まれる `null` データも含めて文字数をカウントできる基本文字列データとして処理することにより、転送しようとする第1の文字列データ中に含まれる `null` データも含めて実効文字数を求めることができる。そして、オペレーティングシステムが転送可能な転送文字列データに第1の文字列データを実効文字数だけ変換することにより、文字列データがどのような型で宣言されていても、文字列データの途中に含まれる「00」の `null` データを文字列データの終端とせず、`null` データも含めて正確に転送文字列データとすることができ

る。このため、標準化された転送文字列データでしかオブジェクト間のデータ転送をサポートできないようなCOMなどのアーキテクチャあるいはその一貫として動作するシステム上であっても、nullデータを含めて正確に文字列データを転送できる。

【0022】

基本文字列データは、文字数データ付きの文字列データ、たとえばCOMでサポートされているBSTR型の文字列データを利用できる。また、オブジェクト間のみならず、アプリケーションプログラムとオブジェクト間におけるデータ転送であっても同様の問題が発生する可能性がある。したがって、第1のオブジェクトは、アプリケーションまたは文字列データを処理可能な周辺装置の種類単位のコントロールオブジェクトであり、第2のオブジェクトは、制御オブジェクトまたは周辺装置毎のサービスオブジェクトである。また、本発明はアプリケーションプログラムに対しても、アプリケーションプログラムを第1のオブジェクトとすれば適用することが可能である。したがって、第1のまた、本明細書において、オブジェクトとはアプリケーションを実行する際にCOMのバインディングメカニズムによりバインディングされるオブジェクトのインスタンスも含まれており、特に記載していないかぎり、オブジェクトとインスタンスを区別していない。

【0023】

このように、本発明の文字列データの処理方法および処理システムによれば、オブジェクトにおける文字列データの宣言された型が文字列データ中にnullデータが含まれないことを前提としたものであっても、その文字列データ中にnullデータが含まれた文字列データをオブジェクト間で正確に転送できる。したがって、従来からのオブジェクトの文字列データの宣言された型を変更しないで、バーコードデータなどのnullデータが含まれる可能性がある文字列データを処理することができる。このため、本発明により、バーコードデータなどを含めた多種多様なデータ形式を正確にハンドリングすることができるオブジェクトベースのPOSシステムを提供することができる。

【0024】

本発明に係る第1のオブジェクトは、アプリケーションプログラムを実行する際に、クライアント側となるオブジェクトプログラムとして提供することが可能である。このオブジェクトプログラムは、当該オブジェクトに対しサーバ側となるオブジェクトに対し、オペレーティングシステムを介して第1の文字列データを転送可能なインターフェイスオブジェクトを呼び出して第1の文字列データを転送する工程を有し、この工程において、当該オブジェクトの第1の文字列データの宣言された型に関わらず前記第1の文字列データを、文字列データ中に含まれるnullデータも含めて文字数をカウントできる基本文字列データとして第1の文字列データの実効文字数を求め、オペレーティングシステムが転送可能な転送文字列データに前記第1の文字列データを前記実効文字数だけ変換するインターフェイスオブジェクトを呼び出す処理を実行可能な命令を有し、コンピュータに読み取り可能な磁気ディスクなどの適当な記録媒体に記録して提供できる。

【0025】

また、本発明に係る第2のオブジェクトも、アプリケーションプログラムを実行する際に、サーバ側となるオブジェクトプログラムとして提供できる。このオブジェクトプログラムは、当該オブジェクトに対しクライアント側となるオブジェクトから、オペレーティングシステム上で稼動するインターフェイスオブジェクトを介して転送された第1の文字列データを、当該オブジェクトにおけ第1の文字列データの宣言された型に関わらず、文字列データ中に含まれるnullデータも含めて文字数をカウントできる基本文字列データとして処理を実行可能な命令を有し、コンピュータに読み取り可能な適当な記録媒体に記録して提供することができる。

【0026】

さらに、本発明に係るインターフェイスオブジェクトは、アプリケーションプログラムを実行する際に、クライアント側となる第1のオブジェクトからサーバ側となる第2のオブジェクトに対し、コンピュータのオペレーティングシステムを介して第1の文字列を転送可能なインターフェイスオブジェクトプログラムとして提供できる。このインターフェイスオブジェクトは、第1のオブジェクトの第1の文字列データの宣言された型に関わらず第1の文字列データを、文字列デ

ータ中に含まれる null データも含めて文字数をカウントできる基本文字列データとして第 1 の文字列データの実効文字数を求め、オペレーティングシステムが転送可能な転送文字列データに第 1 の文字列データを実効文字数だけ変換する処理を実行可能な命令を有し、上記と同様に、コンピュータに読み取り可能な適当な記録媒体に記録して提供できる。

【 0 0 2 7 】

【発明の実施の形態】

以下では、上記にて図 1 および図 2 に基づき説明した分散型オブジェクトで構築される P O S システム 1 0 に基づき本発明についてさらに説明する。まず、本発明を P O S システム 1 0 に適用するために、M F C の C O l e D i s p a t c h D r i v e r クラスから C C o O l e D i s p a t c h D r i v e r クラスのインターフェイスオブジェクトを派生し、そのデータ変換部分を以下のようにオーバーライドする。

```
LPCSTR lpsz = va_arg(argList, LPSTR);
long lLen = ::SysStringByteLen((BSTR)lpsz);
long lDestLen = lLen + 1;
LPWSTR pWideChar = (LPWSTR)CoTaskMemAlloc(lDestLen * 2 * sizeof(WCHAR));
long lTrueLen;
if(lLen) {
    lTrueLen = MultiByteToWideChar(CP__ACP, 0, lpsz, lLen, pWideChar,
    lLen);
}
else {
    lTrueLen = MultiByteToWideChar(CP__ACP, 0, lpsz, 0, pWideChar, lD
estLen);
}
pArg->bstrVal = ::SysAllocStringLen(pWideChar, lTrueLen);
CoTaskMemFree((void*)pWideChar);
if (lpsz != NULL && pArg->bstrVal == NULL)
    AfxThrowMemoryException();
```

```
pArg->vt = VT__BSTR;
```

．．．（９）

上記のデータ変換の処理では、まず、１行目でパラメータリストargListの中から文字列パラメータを取り出し、LPCSTR型としている文字列データlpszにセットする。以下、各行毎に順番に、文字列データlpszをBSTR型にキャストし、文字列データの途中にnullデータがある場合も、それを含めた文字列データの長さ（文字数データ）を求める。さらに、文字数データlLenが０にならないように１を足し、さらに、COMで転送される標準の文字コードである１６ビットのUNICODEに対応してワイドキャラクタに変換するためのメモリを確保する。そして、文字列データlpszを文字数データlLenで長さを指定してワイドキャラクタに変換しLPWSTR型の文字列データpWideCharに格納する。

【００２８】

その後、サービスオブジェクトにCOMを介してデータを渡すために、BSTR型の引数pArgに文字数データlLenで長さを指定して変換する。このようにして変換が終了すると、確保したワイドキャラクタ用のメモリを開放する。また、その下でパラメータが不正な場合は例外を発生させる。そして、サービスオブジェクトに渡す文字列データpArgがBSTR型であることを設定する。

【００２９】

したがって、本例のPOSシステム３０は、システム全体の構成は図１と同様であるが、コントロールオブジェクト３１、インターフェイスオブジェクト３２およびサービスオブジェクト３３においては、図３に示すようにデータが転送される。まず、アプリケーションプログラム１１が、ステップ５１においてバーコードを印刷するためのメソッドであるPrintBarcodeメソッドを呼んで、上記と同様に以下のnullデータ（１６進「００」）を含む文字列データA0を渡すと、コントロールオブジェクト３２が以下のBSTR型の文字列データA1を受け取る。

A0 : 7B 41 00 30 31 32 33 34 35 36 37 38 39 . . . （１）

A1 : 0D 00 00 00 7B 41 00 30 31 32 33 34 35 36 37 38 39 00

．．．（２）

そして、本例のコントロールオブジェクト 3 2 は、ステップ 5 2 において、サービスオブジェクトにバーコードの文字列データを転送するために、C0leDispatchDriverの代わりに、上記のように変換機能をオーバーライドしたインターフェイスオブジェクト 3 8 であるCCo0leDispatchDriverクラスのInvokeHeplerVを呼ぶ。

```
void CCo0leDispatchDriver::InvokeHleperV(DISPID dwDispID, WORD wFlags, VARTYPE vtRet, Void* pvRet, const BYTE*, pbParamInfo, va__list argList)
```

．．． (3)

本例のインターフェイスオブジェクト 3 8 においては、上述したように、ステップ 5 3 で、コントロール 3 2 から渡された文字列データ lpsz が L P C T S T R 型で宣言されているが、B S T R 型にキャストし、文字長さ（文字数）のデータを求める。そして、ステップ 5 4 で C O M を介して転送するために求められた文字数分を欠けることなくワイドキャラクタに変換し、さらに、ステップ 5 5 でワイドキャラクタの B S T R 型の文字列データでサービスオブジェクトに転送する変数 pArg にセットする。したがって、本例のインターフェイスオブジェクト 3 8 では以下のように文字列データが変換される。まず、ステップ 5 4 においては、文字列データの長さを指定して文字列変換関数 MultiByteToWideChar によりワイドキャラクタに変換することにより文字列データ A 1 は次の文字列データ A 2 へに変換される。

```
A 2 :      7B 00 41 00 00 00 30 00 31 00 32 00 33 00 34 00 35 00 36 0
0 37 00 38 00 39 00 . . . ( 5 )
```

上記の文字列データ A 2 に対し、本例では、文字列データの長さを指定しているため、文字列データに含まれる n u l l データ（1 6 進「0 0」）以降のデータもワイドキャラクタに変換される。

【 0 0 3 0 】

さらに、このデータが、ステップ 5 5 において、文字変換マクロ SysAllocStringLen により C O M を介してデータを転送する文字列データの U N I C O D E （1 6 ビット）の B S T R 型の文字列データ A 3 へに変換される。

A 3' : 1A 00 00 00 7B 00 41 00 00 00 30 00 31 00 32 00 33 00 34 0
0 35 00 36 00 37 00 38 00 39 00 . . . (6)'

したがって、サービスオブジェクトには、`null`データを含め、アプリケーションプログラム11から供給された全ての文字列データが渡される。

【 0 0 3 1 】

一方、サービスオブジェクト 33 においては、文字列データを受け取る引数 `pData` が `null` データを終端データとする `Null-Terminal` の `LPCSTR` 型で宣言されるので、以下の文字列データ `A4` を受け取る。

A 4 : 7B 41 00 . . . (7)

したがって、このままLPCTSTR型として文字列データを扱っていると、やはりnullデータ以降のデータを処理することができない。しかしながら、上述したようにCOMを介して文字列データが転送される場合は、メモリ上はBSTR型で転送先のオブジェクトに送られる。したがって、サービスオブジェクトにおいては、その関数のプロトタイプでその引数の型がLPCTSTRと宣言されているだけで、実際のデータはBSTR型である。したがって、このデータの前後のメモリを見ると次のようになっている。

A 5 : 0D 00 00 00 7B 41 00 30 31 32 33 34 35 36 37 38 39 00 . . . (8) ,

したがって、本例のサービスオブジェクト 3 3 においては、以下のように、ステップ 5 6 において、L P C T S T R 型で宣言されている引数 pData を B S T R 型であるとして処理する。

```
lDataLen = SysStringByteLen((BSTR)pData);
```

```
CString strBinaryData (pData, lDataLen);
```

• • • (10)

すなわち、上記の処理では、LPCTSTR型で宣言されている引数pDataをBSTR型にキャストして文字数（文字列長さ）を求め、MFCのCStringクラスの変数strBinaryDataにpDataをその文字数（文字列長さ）IDataLenだけセットしてプリンタから印刷できるようにしている。したがって、アプリケーションプログラム11から供給されたnullコードあるいはデータである16進「00

」を含む文字列がサービスオブジェクト33からnullデータおよびそれ以降のデータが削除されることなくプリンタに伝達され、出力される。

【0032】

このように、本例のコントロールオブジェクト32からサービスオブジェクト38にCOMを介してデータを転送するインターフェイスオブジェクト38はMFCのベースクラスから派生させたものであるが、null-terminateの文字列データLPCTSTR型であっても、その文字列データを文字長さあるいは文字数のデータを含んだ「00」を取り扱えるBSTR型のデータにキャストして本来の有効な文字数（実効文字数）を求め、その文字数に相当するデータを文字列データLPCTSTR型に関わらず、ワイドキャラクタに変換してCOMで転送可能なUNICODEの文字列データにしている。したがって、従来のMFCクラスから派生したインターフェイスオブジェクトでは、LPCTSTR型などのnull-terminateの文字列データとして宣言されている場合は、nullデータ以降がCOMを介してオブジェクト間を転送されるときに削除されてしまうのに対し、nullデータを含む文字列データであっても正確にオブジェクト間で渡すことができる。アプリケーションプログラムを作成する際に、プログラマが独自にマーシャリングを行う代わりにMFCクラスから派生したインターフェイスオブジェクトを使用する場合も同様であり、本発明に係るインターフェイスオブジェクトを利用することによりアプリケーションプログラムからコントロールオブジェクトに対してもnullデータおよびそれ以降のデータが落ちるのを防止できる。

【0033】

また、上記のコントロールオブジェクトおよびサービスオブジェクトにおいては、文字列データは元来のLPCTSTR型で宣言されていても、nullデータに関わる問題を防止できるので、アプリケーションプログラム11に対しユニホームなAPIを提供する分散型オブジェクトによるPOSコントロールシステムの仕様を変更する必要はない。したがって、アプリケーションプログラム11を提供する側、また、それを使用するユーザは、過去のソフトウェア資産を無駄にすることなく本発明によりnullデータも正確に取り扱うことができるPO

Sコントロールシステム39を利用することができる。

【0034】

さらに、従来のnull-terminateの文字列データをそのままnull-terminateとして変換してしまうインターフェイスオブジェクトで「00」を含んだ文字列データを強引に取り扱おうとすると、バイナリコンバージョンを利用して「00」を強制的に他の値に変換してからオブジェクト間を転送し、オブジェクト内で元に戻すという方法がないわけではない。しかしながら、処理が複雑になることはもちろんであるが、オブジェクト内で、他のオブジェクトの変換処理とマッチした処理を行う必要があり、転送元のオブジェクトの仕様に依存することになる。したがって、ソフトウェアの部品化と再利用の可能性を高めるというオブジェクト指向プログラムとは相反するものとなるので採用しがたい。さらに、バイナリコンバージョンされたデータを処理できないオブジェクトであれば、異なった文字列を出力する可能性がある。これに対し、上記に説明した本発明の方法であれば、nullデータおよびそれ以降のデータをそのままの状態での次のオブジェクトに転送できるので、それを受け取ったオブジェクトはnull-terminateの文字列データとして処理しても良く、あるいは上記のようにBSTR型にキャストして「00」を含めて処理しても良い。このため、従来のオブジェクトに影響を及ぼすこともなく、nullデータを取り扱うことができるコントロールシステムを構築できる。

【0035】

本例のPOSコントロールシステム39を構築するコントロールオブジェクト32、サービスオブジェクト33およびインターフェイスオブジェクト38は、図3に示した各ステップの処理を実行可能な命令を有するプログラムとしてCD-ROMあるいはその他のコンピュータに読み取り可能な記録媒体に記録して提供することができる。そして、コンピュータ内のOS14により管理されるメモリ領域にインストールすることにより、各々のオブジェクトのメソッドあるいはプロパティなどが呼び出されると、各々のオブジェクトのインスタンスが生成され、バインドされてアプリケーションプログラムの所望の処理が実行される。

【0036】

なお、以上では、バーコードなどの多種多様なデータを取り扱う可能性があるために `null` データを実際に処理する可能性が高い P O S システムを例に本発明を説明しているが、文字列データを取り扱う他のシステムにおいても本発明を同様に適用できる。

【 0 0 3 7 】

【発明の効果】

以上に説明したように、本発明においてはバイナリコンバージョンを利用して文字列データを特殊な形式に変換することなく `null` データ（16進「00」）を途中に含む文字列データであっても正確にオブジェクト間を転送できる。したがって、バーコードデータなどの `null` データを含む可能性のある文字列データを処理可能な方法および装置をオブジェクトサービスによって提供することが可能となり、P O S システムなどの多種多様なデータを取り扱う可能性のあるシステムを容易に構築できるソフトウェア開発手段を提供できる。

【図面の簡単な説明】

【図 1】

オブジェクトサービスにより P O S システムの概要を示す図である。

【図 2】

アプリケーションプログラムおよびオブジェクト間を文字列データが転送する様子を示す図である。

【図 3】

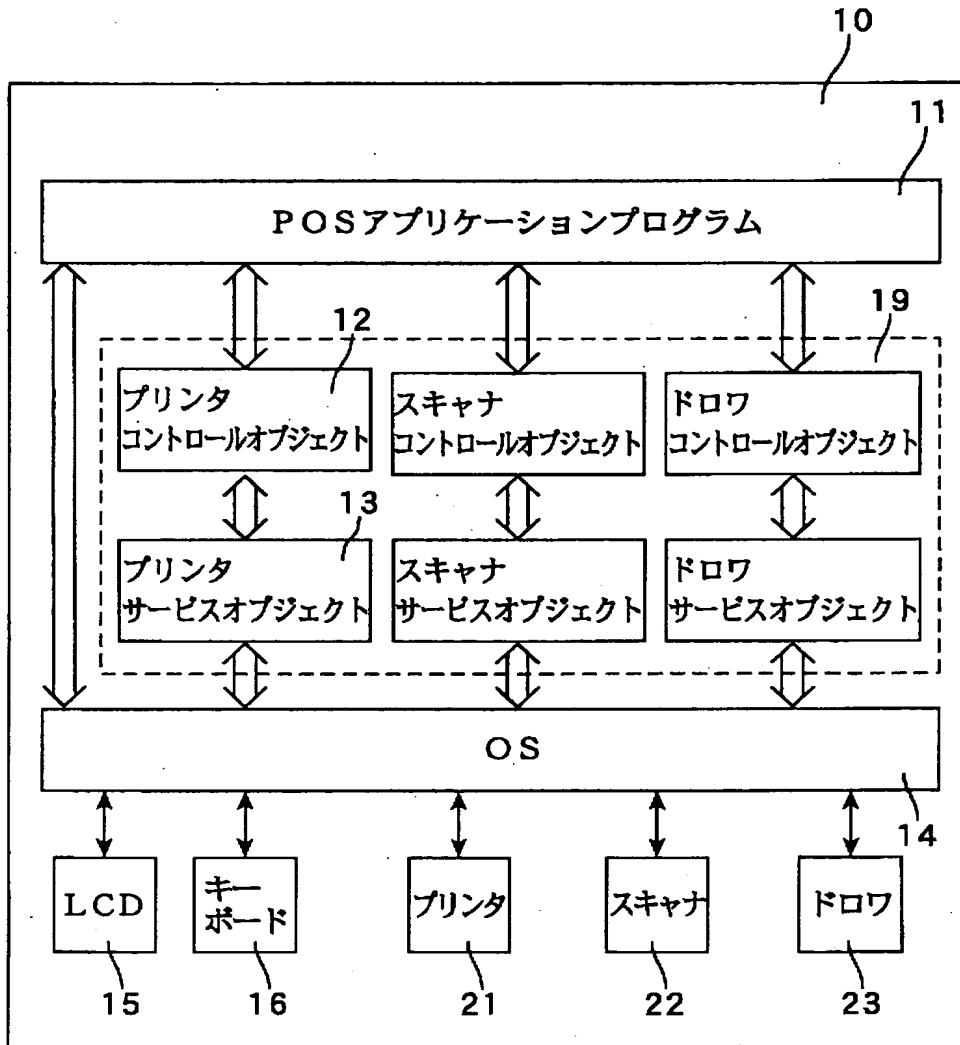
本発明のコントロールオブジェクト、インターフェイスオブジェクトおよびサービスオブジェクトの処理の概要を示す図である。

【符号の説明】

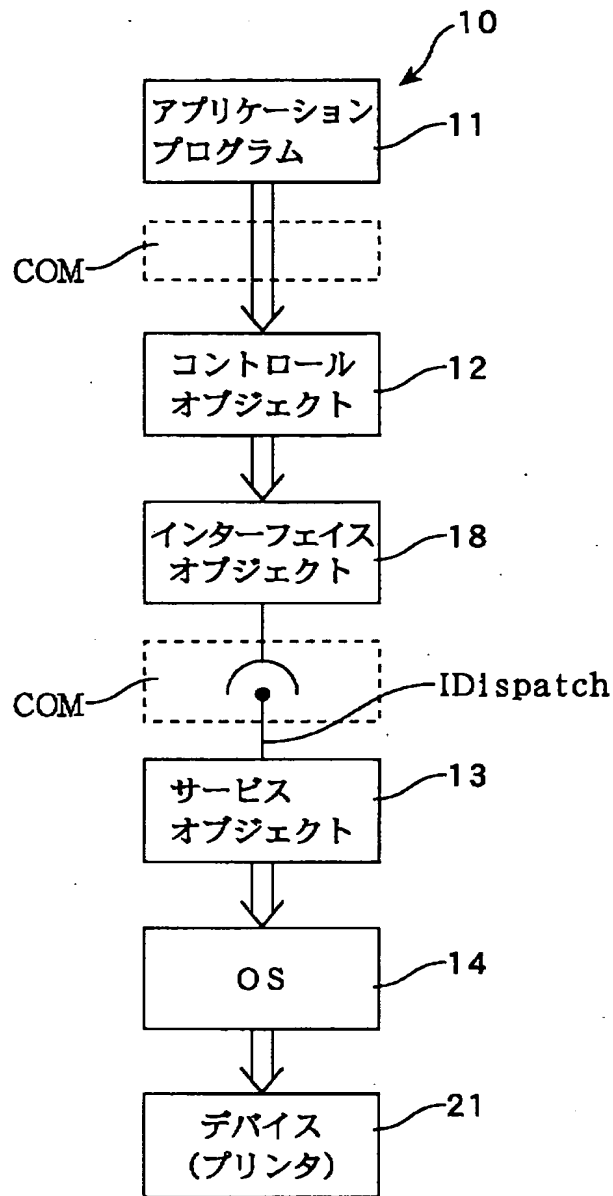
- 1 0 P O S システム
- 1 1 アプリケーションプログラム
- 1 2、3 2 コントロールオブジェクト
- 1 3、3 3 サービスオブジェクト
- 1 8、3 8 インターフェイスオブジェクト
- 1 9、3 9 P O S コントロールシステム（オブジェクトサービス）

【書類名】 図面

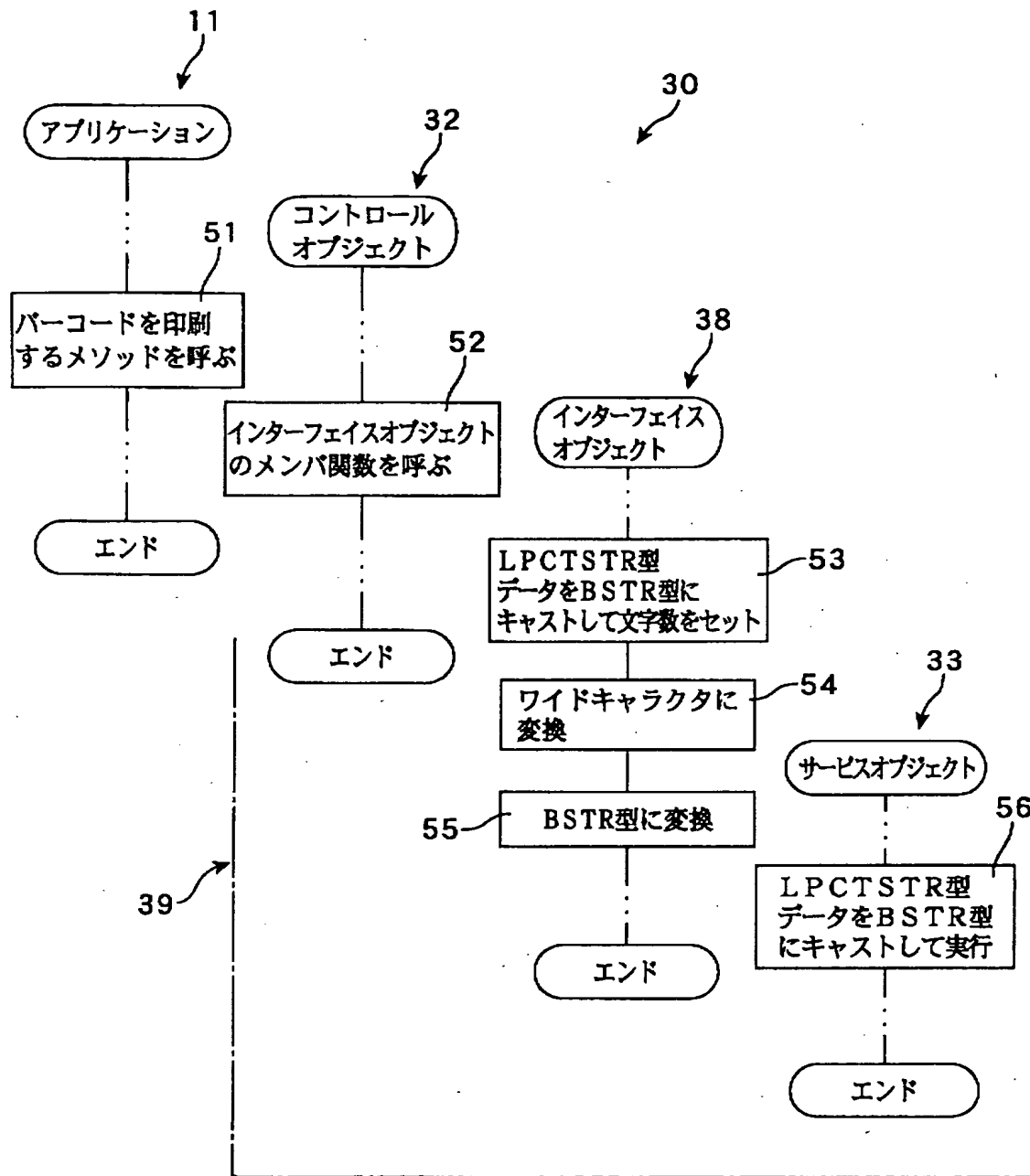
【図 1】



【図 2】



【図 3】



【書類名】 要約書

【要約】

【課題】 コントロールオブジェクトあるいはサービスオブジェクトの文字列データの型の宣言を変更したり、複雑な処理を行わないで、`null`データを含む文字列データを正確にオブジェクト間で転送できるようにする。

【解決手段】 コントロールオブジェクト 3 2 からサービスオブジェクト 3 3 に文字列データを渡すインターフェイスオブジェクト 3 8 において、コントロール 3 2 から渡された文字列データが `LPCTSTR` 型であっても `BSTR` 型にキャストし、文字長さ（文字数）のデータを求め、その文字数分を欠けることなくワイドキャラクタに変換し、さらに、`COM` を介して転送するためにワイドキャラクタの `BSTR` 型の文字列データにセットする。これにより `null` データおよびそれ以降の文字列も正確にサービスオブジェクト 3 3 に渡すことができる。

【選択図】 図 3

出 願 人 履 歴 情 報

識別番号 [000002369]

1. 変更年月日 1990年 8月20日
[変更理由] 新規登録
住 所 東京都新宿区西新宿2丁目4番1号
氏 名 セイコーエプソン株式会社